

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

Claim 1 (canceled).

Claim 2 (Currently Amended): A computer implemented method for mapping storing objects ~~onto~~ to a lightweight directory access protocol repository, comprising:

dynamically determining persistent attributes associated with an object in response to a request to store the object in a lightweight directory access protocol ("LDAP") repository, wherein the object includes attributes, the persistent attributes are a subset of the attributes, and the persistent attributes each comprise a persistent attribute value;

determining a path from information in the object, wherein the path identifies a location in the LDAP repository to store the object and the path identifies a LDAP object that corresponds to the object;

retrieving one or more persistent attribute values associated with the object;  
and

storing the object in the LDAP repository so that the persistent attributes are stored in a format that is useable by other applications ~~other than the object-oriented programming application~~, wherein storing the object in the LDAP repository comprises:

mapping the persistent attributes to LDAP attributes, wherein the LDAP attributes are associated with the LDAP object identified by the determined path;

populating the LDAP attributes with the one or more persistent attribute values; and

passing the LDAP attributes populated with the one or more persistent attribute values to the LDAP repository.

Claim 3 (original): The method of claim 2, wherein the persistent attributes each have a name and wherein mapping the persistent attributes to LDAP attributes comprises adding a prefix to the persistent attribute name.

Claim 4 (previously presented): The method of claim 3, wherein the prefix identifies an object-oriented programming application and an organization.

Claim 5 (previously presented): The method of claim 2, wherein the one or more persistent attribute values are passed to the LDAP repository as an LDAP object comprising the LDAP attributes and the one or more persistent attribute values.

Claim 6 (previously presented): The method of claim 2, wherein the object is used in an object-oriented programming application and the object-oriented programming application has a name and the object has a name and wherein the path includes the object-oriented programming application name, a container name and the object name.

Claim 7 (previously presented): The method of claim 2, wherein the object represents one of the following: a user, a node, a node group, a role or a tool.

Claim 8 (previously presented): The method of claim 2, wherein the objects are Java objects.

Claim 9 (previously presented): The method of claim 6, wherein the object-oriented programming application is implemented in Java.

Claim 10 (previously presented): The method of claim 9, wherein the one or more persistent attribute values are retrieved from the object using Java reflection.

Claims 11-15 (canceled).

Claim 16 (currently amended): A computer readable storage medium containing instructions for ~~mapping~~ storing objects ~~onto~~ to a lightweight directory access protocol repository, by:

dynamically determining persistent attributes associated with an object in response to a request to store the object in a lightweight directory access protocol ("LDAP") repository, wherein the object includes attributes, the persistent attributes are a subset of the attributes, and the persistent attributes each comprise a persistent attribute value;

determining a path from information in the object, wherein the path identifies a location in the LDAP repository to store the object and the path identifies a LDAP object that corresponds to the object;

retrieving one or more persistent attribute values associated with the object;  
and

storing the object in the LDAP repository so that the persistent attributes are stored in a format that is useable ~~to by other applications other than the object-oriented programming application~~, wherein storing the object in the LDAP repository comprises:

mapping the persistent attributes to LDAP attributes, wherein the LDAP attributes are associated with the LDAP object identified by the determined path;

populating the LDAP attributes with the one or more persistent attribute values; and

passing the LDAP attributes populated with the one or more persistent attribute values to the LDAP repository.

Claim 17 (currently amended): The computer readable storage medium of claim 16, wherein the objects are Java objects.

Claim 18 (currently amended): The computer readable storage medium of claim 16, wherein the object is used in an object-oriented programming application and the object-oriented programming application is implemented in Java and the one or more persistent attribute values are retrieved from the object using Java reflection.

Claim 19 (canceled).

Claim 20 (currently amended): A computer system that supports ~~mapping~~ storing objects ~~onto~~ to a lightweight directory access protocol repository, comprising:

a lightweight directory access protocol ("LDAP") repository;

a processor that runs an object-orient programming application, wherein the object-oriented programming application generates:

an object, wherein the object includes attributes and is used in an object-oriented programming application;

a persistent data manager, that acts as a layer between the object and the LDAP repository, wherein the persistent data manager stores the object in the LDAP repository by:

dynamically determining persistent attributes associated with the object in response to a request to store the object in the LDAP repository, wherein the persistent attributes are a subset of the attributes and the persistent attributes each comprise a persistent attribute value;

determining a path from information in the object, wherein the path identifies a location in the LDAP repository to store the object and the path identifies a LDAP object that corresponds to the object;

retrieving one or more persistent attribute values associated with the object; and

storing the object in the LDAP repository so that the persistent attributes are stored in a format that is useable ~~to by other applications other than the object-oriented programming application~~, wherein storing the object in the LDAP repository comprises:

mapping the persistent attributes to LDAP attributes, wherein the LDAP attributes are associated with the LDAP object identified by the determined path;

populating the LDAP attributes with the one or more persistent attribute values; and

passing the LDAP attributes populated with the one or more persistent attribute values to the LDAP repository.